PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06T 11/40 | A1

(11) International Publication Number: WO 00/33256

(43) International Publication Date: 8 June 2000 (08.06.00)

(21) International Application Number: PCT/US99/28003

(22) International Filing Date: 23 November 1999 (23.11.99)

(30) Priority Data:
09/201,070 30 November 1998 (30.11.98) US

(71) Applicant: QUANTUM3D, INC. [US/US]; 6810 Santa Teresa Boulevard, San Jose, CA 95119 (US).

(72) Inventor: HUELSON, Phillip, W.; 1580 Neston Way, Los Altos, CA 94024 (US).

(74) Agent: HICKMAN, Paul, L.; Hickman Stephens Coleman & Hughes, LLP, P.O. Box 52037, Palo Alto, CA 94303 (US).

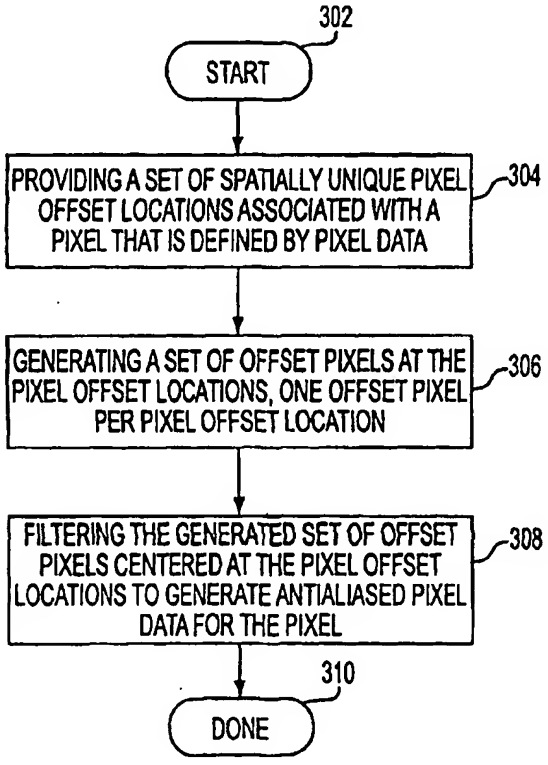
(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: PIXEL OFFSET ANTIALIASING

(57) Abstract

A method and system for antialiasing a pixel using offset pixel data. The method includes providing a set of pixel offset locations for the pixel. Each of the pixel offset locations is provided relative to a center of the pixel and each of the pixel offset location is unique from every other pixel offset locations (element 304). The method also includes generating a set of offset pixels at the pixel offset locations associated with the pixel with one offset pixel per pixel offset location (element 306). Each of the offset pixels is defined by offset pixel data. In addition, the method includes filtering the set of offset pixels at the pixel offset locations to generate antialiased pixel data for the pixel (element 308).



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

PIXEL OFFSET ANTIALIASING

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates generally to antialiasing, and more particularly to antialiasing in real time using multiple concurrent streams of offset pixel data.

2. Description of the Related Art

The root cause of aliasing in computer graphics is the creation of images by a regular sampling process in the space domain; and the case of animation and/or realtime applications, in the time domain. Familiar manifestations of aliasing are jagged lines, and jagged silhouette edges of a surface unit. (A silhouette edge is a boundary that exhibits high luminance contrast against its background.) One particularly degenerate manifestation of aliasing occurs when texture is used on the surface of an object subject to a perspective transformation. In this case, the texture in the distance loses its form and produces highly noticeable low-frequency aliases and moire' interference patterns. Another aliasing artifact occurs when small objects or very thin lines, whose spatial extent or width is less than the area of a pixel, are rendered or not depending on whether they are intersected by a sample point. These artifacts are particularly troublesome in animated and realtime sequences in which jagged edges or lines 'crawl' (move spatially as they intersect different display sample points) and small objects or thin lines (or portions of a thin line) may 'scintillate' (appear and disappear).

Three established methods of removing or reducing aliasing artifacts in computer graphics include: supersampling (also known as postfiltering); prefiltering (also known as area sampling); and stochastically sampling. Supersampling methods render an image at higher resolution than the final resolution to be displayed. For example, an image may be rendered at a resolution that is finer than a pixel by rendering multiple subpixels for each pixel to be displayed. The subpixels are then combined to generate a final pixel for display. For example, a virtual image can be rendered at four times the final rendering resolution using normal rendering techniques for shading and hidden surface removal. In this example, groups of 4 X 4 'superpixels' would then be reduced to a single pixel by weighting each superpixel value by a filter weight, summing, and normalizing. This method works well with many computer graphic images and is easily integrated into a Z-buffer algorithm, but it does not work well with images whose spectrum energy does not fall off with increasing frequency such as textures rendered in

perspective. Space variant filters, in which both the weights of the filter kernel and its shape are varied, have been used on perspective corrected textures to improve supersampled antialiasing results. Supersampling methods differ in minor ways through the multiples of the final display resolution used, and in the shape of the filter(s) applied.

In prefiltering or area sampling antialiasing methods, the algorithm performs subpixel geometry in the continuous image domain and returns for each pixel an intensity which is computed by using the areas of visible subpixel fragments as weights in an intensity sum for that pixel. One method of prefiltering is to compute the intensity of a pixel by clipping polygons against a square pixel boundary. If polygon fragments overlap within a square pixel boundary they are sorted in Z (depth) and clipped against each other to produce visible fragments. A final intensity is computed by multiplying the shade of a polygon by the area of its visible fragment and summing. This method is generally deemed prohibitively computationally intensive. Further developments in prefiltering have involved approximating subpixel fragments with bit masks, combined with a Z-buffer, to create a an anti-aliased, area-averaged, accumulation buffer, also known as the 'A-buffer.' In this technique, coverage and area weighting are accomplished by using bitwise logical operators between the bit patterns or masks using polygon fragments. Still other developments in antialiasing through area sampling take advantage of the fact that the way in which a polygon can cover a pixel can be approximated by a limited number of cases. Therefore contributions to the convolution integral can be precomputed and stored in look-up tables indexed by polygon fragments. It should be noted that the convolution is not restricted to a single pixel extent, but rather can extend over, for example, a 3 x 3 output pixel area. Pixels act as accumulators, whose final value is correct when all fragments that can influence its value have been taken into account.

Stochastic sampling has been used to provide a general form of antialiasing solution. The basis of this method is to perturb the position of the sampling points (jitter), thereby mapping high-frequency aliasing artifacts above the Nyquist limit into noise. The frequency of the information being sampled relative to the sample frequency determines the energy of the noise, while the distribution of the perturbations determine the spectral character of the noise. Generally stochastic sampling can been described as a two stage process: 1) sample the image using a sampling grid where the (x,y) position of each sampling point has been subject to a random perturbation; 2) use these sample values with a reconstruction filter to determine the pixel intensities to which the unperturbed sample positions correspond.

However, the conventional antialiasing techniques present several disadvantages. For example, an inevitable side-effect of virtually all forms of antialias filtering is blurring. Blurring occurs because information is integrated from a number of neighboring pixel or subpixel elements. Hence, the choice of the spatial extent of the filter may be a compromise. A wide filter will blur the image more than a narrow filter. The narrow filter will however exhibit a higher cut-off frequency and thus increased aliasing artifacts when compared to the lower cut-off frequency and reduced aliasing artifacts offered by a wide filter.

In the case of supersampling, this method is not suitable for dealing with very small objects or very narrow lines as these images tend to scintillate. It does not work well with textures rendered in perspective because their spectrum energy does not fall off with increasing frequency. The computation required for supersampling is not context dependent, therefore a scene that exhibits a few large polygons is subject to the same computational overhead as one with a large number of small area polygons.

Furthermore, the memory requirements for supersampling are typically large. If the method is to be used with a Z-buffer, the supersampled image has to be created and stored before the filtering process can be applied. Supersampling increases the storage requirements of the Z-buffer by a multiple of the increase in effective area in the supersample (e.g. for a 4 x 4 supersample, the Z-buffer size is increased by a factor of 16).

In the case of prefiltering or area sampling, the severe computational overhead alone inherent in the method is often considered prohibitive. On the other hand, stochastic sampling suffers from difficulty in implementation, and display artifacts produced by the intentional substitution of white noise for alias artifacts as previously discussed. Two-dimensional sampling perturbations are difficult to incorporate into 'standard' image synthesis methods because the algorithms for image synthesis are founded on uniform incremental methods in screen space. Therefore common image generation systems are not capable of generating random perturbations required to implement stochastic sampling. Separating image creation into the two phases of generation in a continuous domain followed stochastic sampling has been suggested, but in practice generation and sampling are not easily decoupled.

The conventional antialiasing methods mentioned above also require a relatively high degree of computational effort. The high degree of computational effort is sometimes combined with dedicated hardware acceleration to enhance performance in areas including, for example: multiplication and accumulation, Z-buffering, clipping, bitwise logical

operators, and table look-ups. Even with dedicated hardware, its is extremely difficult to render high resolution antialiased images in real time (for example, 30 to 60 frames per second at output resolutions of 1024x768 and above) for use in applications such as animation or visual simulation using any of the aforementioned techniques.

Thus, what is needed is a method and a system for efficiently antialiasing pixel data to generate quality medium- to high-resolution images without employing a costly computational efforts or large amounts of specialized hardware. Specifically, the system and method should reduce or eliminate: jagged and crawling effects of lines and silhouette edges; scintillation effects on small objects and very thin lines; and low-frequency aliases and Moire interference patterns in textures. Additional enhancements could include filtering that intensifies thin off-axis lines to offset the effect of longer distances between points on off-axis lines. What is further needed is a system and a method that perform such antialiasing without substantial delay or video bandwidth limitations so as to meet real time requirements of performance critical applications.

SUMMARY OF THE INVENTION

The present invention fills these needs by providing a method and a system for antialiasing a pixel using offset pixel data from multiple simultaneous pixel data streams. It should be appreciated that the present invention can be implemented in numerous ways, including as a process, an apparatus, a system, a device, a method, or a computer readable medium.

In one embodiment, the present invention provides a method, in a computer system, for antialiasing a pixel defined by pixel data. The method includes providing a set of pixel offset locations for the pixel. Each of the pixel offset locations is provided relative to a center of the pixel and each of the pixel offset locations is unique from every other pixel offset locations. The method also includes generating a set of offset pixels at the pixel offset locations associated with the pixel with one offset pixel per pixel offset location. Each of the offset pixels is defined by offset pixel data. In addition, the method includes filtering the set of offset pixels at the pixel offset locations to generate antialiased pixel data for the pixel.

In another embodiment, a system for antialiasing a pixel defined by pixel data is disclosed. The system includes a bus, a processor coupled to the bus, a memory coupled to the bus, and a storage device coupled to the bus. The system also includes a set of graphics subsystems coupled to the bus. The set of graphics subsystems is associated with a set of pixel offset locations for the pixel. Each of the pixel offset locations is provided relative to a center of the pixel and each of the pixel offset locations is unique from every other pixel offset locations. The set of graphics subsystems is arranged to generate a set of offset pixels at the pixel offset locations associated with the pixel with one offset pixel per pixel offset location. Each of the offset pixels is defined by offset pixel data. The system further includes an antialiasing unit coupled to receive the set of offset pixel data to filter the set of offset pixels at the pixel offset locations to generate antialiased pixel data for the pixel.

In yet another embodiment, the present invention provides a system for antialiasing a pixel defined by pixel data. The system includes means for generating a set of offset pixels at a set of pixel offset locations with one offset pixel per pixel offset location. Each of the offset pixels is defined by offset pixel data. The generating means is associated with the set of pixel offset locations for the pixel. Each of the pixel offset locations is provided relative to a center of the pixel and each of the pixel offset locations is unique from every other pixel offset locations.

The system also includes antialiasing means for filtering the set of offset pixels at the pixel offset locations to generate antialiased pixel data for the pixel.

Advantageously, the present invention generates antialiased pixel data at substantially real time without the computationally intensive tasks and delay associated with conventional antialiasing techniques. By associating a set of graphics subsystems with a set of pixel offset locations to generate multiple simultaneous offset pixel data streams, which in turn are used to produce antialiased pixel data, the present invention also provides the flexibility and scalability required to adapt both the filter technique and the image resolutions. Careful selection of the pixel offset locations creates the foundation for the development of a image within substantially reduced aliasing artifacts, and the offset locations are easily adjusted under software control to change the selected filtering. Further, the present invention provides an antialiasing unit that employs only simple adders as the antialiasing computation unit(s), with no inherent accumulation required or accumulation delay (other than implementation-dependent pipeline delays inherent many high-speed digital systems). These adders generate antialiased pixel data by averaging the input offset pixel data using a set of adders adapted to add m-bit input R, G, and B components of the offset pixel data to generate (m+1) bit output R, G, B components. For N offset pixel data inputs of m-bit each, the antialiasing unit may generate ($m+\log_2 N$) bit antialiased pixel data. This arrangement allows loss-less antialiasing in real time even at a substantially high resolution. These and other advantages of the present invention will become apparent to those skilled in the art upon a study of the specification and drawings describing the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and form a part of this specification, illustrate embodiments of the invention and, together with the description, serve to explain the principles of the invention.

Figure 1A illustrates a block diagram of a computer graphics system within which the present invention may be implemented or practiced.

Figure 1B illustrates a block diagram of the computer graphics system within which the present invention may be implemented or practiced in accordance with another embodiment of the present invention.

Figure 2 shows four graphics subsystems coupled to the antialiasing unit for performing pixel offset antialiasing in accordance with one embodiment of the present invention.

Figure 3A illustrates a flowchart of a method for antialiasing a pixel in accordance with one embodiment of the present invention.

Figure 3B shows a more detailed flow chart of an operation of providing a set of spatially unique pixel offset locations in accordance with one embodiment of the present invention.

Figure 4 illustrates an exemplary pixel offset location pattern and offset pixels for antialiasing a pixel.

Figures 5A through 5H illustrate exemplary pixel offset location patterns having four pixel offset locations that are defined in an (x,y) coordinate system for antialiasing a pixel.

Figure 6A illustrates a pixel offset location pattern having eight pixel offset locations.

Figure 6B shows another exemplary pixel offset location pattern having six pixel offset locations.

Figures 7A through 7D illustrate square and diamond pixel offset location patterns in accordance with certain embodiments of the present invention.

Figures 8A through 8D illustrate triangular pixel offset location patterns in accordance with some embodiments of the present invention.

Figures 9A and 9B show exemplary pixel offset location patterns having five pixel offset locations.

Figures 10A and 10B illustrate exemplary pixel offset location patterns having seven pixel offset locations.

Figure 11 shows a flow chart of a method performed by an antialiasing unit in accordance with one embodiment of the present invention.

Figure 12 illustrates a more detailed schematic diagram of the antialiasing unit for four graphics subsystems in accordance with one embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention will now be described in detail with reference to exemplary preferred embodiments as illustrated in the accompanying drawings. In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art, that the present invention can be practiced without some or all of these specific details. In other instances, well known process steps have not been described in detail in order not to unnecessarily obscure the present invention.

The present invention provides methods and systems for efficiently antialiasing pixel data to generate quality medium- to high-resolution images without employing a costly computational efforts or large amounts of specialized hardware. For example, the system and method reduce or eliminate jagged and crawling effects of lines and silhouette edges, scintillation effects on small objects and very thin lines, and low-frequency aliases and Moire interference patterns in textures. An additional enhancement includes filtering that intensifies thin off-axis lines to offset the effect of longer distances between points on off-axis lines. Additionally the system and method of the present invention allow antialiasing without substantial delay or video bandwidth limitations so as to meet real time requirements of performance critical applications.

The method and system for antialiasing a pixel generates a set of offset pixel data at a set of locations that are spatially offset from the pixel in accordance with some embodiments of the present invention. The set of offset pixel data is then averaged substantially in real time to generate antialiased pixel value for the pixel. The present invention also provides a method, system, and a device for antialiasing a pixel in real time without an accumulation buffer.

Figure 1A illustrates a block diagram of a computer graphics system 100 within which the present invention may be implemented or practiced. It should be appreciated that the computer graphics system 100 is exemplary only and that the present invention can operate within a number of different computer system configurations including general purpose computer systems, embedded computer systems, and computer systems that employ multiple processors and/or multiple peripheral and/or graphics buses. In describing various embodiments of the present invention, certain processes and operations are realized as a series of instructions

(e.g., software programs) that reside within computer readable memory units of computer graphics system 100 and are executed by processors therein.

The computer graphics system 100 may be any computer-controlled graphics system suitable for generating 2D or 3D images. The computer graphics system 100 includes one or more bus 102 for transmitting digital information between the various parts of the computer system. One or more processors 104 for processing information are coupled to one or more bus 102. The information, together with the instructions for processing the information, are stored in a hierarchical memory system comprised of a mass storage device 108, a read only memory (ROM) 110, and a main memory 106. The mass storage device 108 is used to store a vast amount of data and may include one or more hard disk drives, floppy disk drives, optical disk drives, tape drives, CD-ROM drives, or any number of other types of storage devices having media for storing data digitally. The ROM 110 is used to store digital data on a permanent basis, such as instructions for the microprocessors. The main memory 106 is used for storing digital data on an intermediate basis. The main memory 106 can be DRAM, SDRAM, RDRAM, or any other suitable memory for storing data while the computer graphics system 100 is turned on. The one or more processors 104, one or more buses 102, the mass storage device 108, and the ROM 110 together define a computer subsystem 130.

The computer system 100 also includes a set of graphics subsystems 112 coupled to one or more bus 102 for processing graphics data in accordance with one embodiment of the present invention. The computer system may include N graphics subsystems 112 from 0 to (N-1). Each of the graphics subsystems 112 is arranged to process graphics data to generate an offset pixel at a unique location that is spatially offset from a pixel to be antialiased and displayed. In this arrangement, the processor 104 provides the graphics subsystems 112 with the graphics data, such as drawing commands, coordinate vertex data, and other data related to an object's geometric position, color, texture, shading, and other surface parameters. Preferably, the graphics data includes spatial pixel offset values to allow the graphics subsystems 112 to generate offset pixel data.

From the graphics data provided, the graphics subsystems 112 generate intermediate offset pixel data at each of the assigned pixel offset location. The graphics subsystems 112 convert the graphics data into a set of offset pixel data at the spatially offset locations. An antialiasing unit 114 is coupled to the graphics subsystems to receive the set of offset pixel data for filtering. Specifically, the antialiasing unit 114 generates an average pixel value from the set of offset pixel data as an antialiased pixel value. The antialiasing unit 114 also communicates

synchronization signals such as pixel clock signal, VSYNC, HSYNC, and blanking signal with the graphics subsystems 112 for synchronization.

The antialiasing unit 114 transmits the antialiased pixel value to a digital-to-analog converter (DAC) 116, which converts the pixel value into analog RGB signals. The analog signals are then fed into a display device 118 for display. It should be noted that the computer graphics system 100 processes a stream of graphics data to antialias a stream of pixel data to display a frame of pixels in substantially real time.

Several other devices may also be coupled to the computer graphics system 100. For example, an alphanumeric keyboard 120 may be used for inputting commands and other information to processor 104 via the bus 102. A user input device such as a cursor control device 122 (e.g., a mouse, trackball, joystick, and touchpad) may also be used for positioning a movable cursor and selecting objects on a computer screen.

The computer graphics system 100 may be implemented using a plurality of processors 104, a plurality of buses 102, and a plurality of graphics subsystems 112. For example, Figure 1B illustrates a block diagram of the computer graphics system 100 within which the present invention may be implemented or practiced in accordance with another embodiment of the present invention. The computer graphics system 100 includes a plurality of computer subsystems 130, each of which is coupled to a bus 102. In turn, each of the buses 102 are coupled to one or more graphics subsystems 112.

Figure 2 shows four graphics subsystems 202, 204, 206, and 208 coupled to the antialiasing unit 114 for performing pixel offset antialiasing in accordance with one embodiment of the present invention. Each of the graphics subsystems 202, 204, 206, and 208 includes a graphics processor GP0 210, GP1 214, GP2 218, and GP3 222, respectively, and a frame buffer FB0 212, FB1 216, FB2 220, and FB3 224, respectively. Each of the graphics processors GP0 210, GP1 214, GP2 218, and GP3 222 receives graphics data and uses the respective frame buffer FB0 212, FB1 216, FB2 220, and FB3 224, respectively, in performing various well known graphics operations such as geometry transformation, scan conversion, and rasterization to produce offset pixel values at the assigned pixel offset locations.

For example, the graphics processors GP0 210, GP1 214, GP2 218, and GP3 222 convert graphical data into a screen coordinate system and may perform projection and transformation processes to generate graphics primitives such as points, lines, polygons, polyhedra, and the like. The graphics processors GP0 210, GP1 214, GP2 218, and GP3 222 may generate offset pixel

data based on the received primitives by interpolating straight lines so that each intermediate value need not be individually and separately computed. The graphics processors GP0 210, GP1 214, GP2 218, and GP3 222 may also perform additional graphics functions such as Z-buffering, blending, and texturing on the offset pixel data. The resulting pixel data values (e.g., RGB values) may then be stored in the frame buffers FB0 212, FB1 216, FB2 220, and FB3 224. The graphics processors GP0 210, GP1 214, GP2 218, and GP3 222 may be implemented by using any processor suitable for generating pixel data from graphics data. For example, Voodoo2™ and Banshee2™ processor from 3Dfx, Inc. of San Jose, California, may be used for any of the graphics processors GP0 210, GP1 214, GP2 218, and GP3 222.

In order to synchronize the processing of offset pixel data, each of the graphics subsystems 202, 204, 206, and 208 receives a master pixel clock PCLK, HSYNC, and VSYNC signals from the antialiasing unit 14. The horizontal sync signal HSYNC indicates the beginning of a new scan line while the vertical sync signal VSYNC specifies the top of an image frame. The pixel clock PCLK indicates each pixel in a horizontal scan line.

The graphics subsystems 202, 204, 206, and 208 also include a clock generator CG0 226, CG1 228, CG2 230, and CG3 232, respectively. Each of the clock generators CG0 226, CG1 228, CG2 230, and CG3 232, generates its own pixel clock PCLK0, PCLK1, PCLK2, and PCLK3, respectively, and blanking signals. The generated pixel clock signals and one of the blanking signals are then provided to the antialiasing unit 114. A blanking signal is typically used to bring down a video signal to a display device when a scan line moves from right edge of a screen down to a left edge of the screen. Any one of the blanking signals may be selected to provide a reference blanking signal.

The antialiasing unit 114 generates a pixel clock PCLK, HSYNC, and VSYNC signals and feeds the signals to the graphics subsystems 202, 204, 206, and 208. The antialiasing unit 114 is coupled to the graphics subsystems 202, 204, 206, and 208 to receive the offset pixel data, the pixel clock signals, and the selected blanking signal. One of the pixel clocks from the graphics subsystems 202, 204, 206, and 208 is selected as a master pixel clock. Any one of the pixel clocks may be selected. In accordance with one embodiment of the present invention, the antialiasing unit 206 performs filtering by averaging the input offset pixel data from the graphics subsystems 202, 204, 206, and 208 to generate antialiased pixel data value. The antialiased pixel data value is then transmitted to the DAC 116 for conversion into analog RGB signals.

Figure 3A illustrates a flowchart of a method for antialiasing a pixel in accordance with one embodiment of the present invention. The pixel is defined by pixel data such as RGB

values. The method starts in operation 302 and proceeds to operation 304, where a set of spatially unique pixel offset locations are provided for the pixel. The spatially unique pixel offset locations define an offset pattern around the center of the pixel. Each of the pixel offset locations are provided relative to a center of the pixel and each of the pixel offset locations are spatially unique from every other pixel offset locations. Then in operation 306, a set of offset pixels are generated at the pixel offset locations. One offset pixel is generated at each pixel offset location. After generating a set of offset pixels associated with the pixel, the generated set of offset pixels are filtered to generate antialiased pixel data for the pixel. In the filtering operation, the set of offset pixels associated with the set of pixel offset locations are averaged, preferably by using a series of add operations. The method then terminates in operation 310.

Figure 3B shows a more detailed flow chart of the operation 304 of providing a set of spatially unique pixel offset locations in accordance with one embodiment of the present invention. The operation 304 starts in operation 320 and proceeds to operation 322, where a set of graphics subsystems are provided. Preferably, N graphics subsystems are provided to generate N offset pixel data.

Then in operation 324, the set of graphics subsystems are associated with the set of offset pixels located at the associated pixel offset locations. One graphics subsystem is preferably associated with one unique pixel offset location. The operation 304 then terminates in operation 326.

For efficient antialiasing, a variety of patterns of pixel offset locations may be defined in a coordinate system for generating offset pixels at the associated offset locations. Figure 4 illustrates an exemplary pixel offset location pattern 400 and offset pixels 404, 406, 408, and 410 for antialiasing a pixel 402. The pattern 400 contains four (e.g., N=4) pixel offset locations 414, 416, 418, and 420. The pixel 402 is centered at an origin (0,0) 402 in an (x,y) coordinate. The offset pixels 404, 406, 408, and 410 include pixel offset locations 414, 416, 418, and 420, respectively. Each of the pixel offset locations 414, 416, 418, and 420 coincides preferably with a center of the associated offset pixel.

It should be noted that all the offset pixels 404, 406, 408, and 410 and the pixel 412 are defined and lie within a set of four quadrants of the (x,y) coordinate system: (+,+), (+,-), (-,-), and (+,-), which, in turn, define an (N/2)x(N/2) pixel block. For example, for N=4, the quadrants define a 2x2 pixel block. Although such a pixel block is illustrated, the present invention may employ pixel block configuration including 3x3, 4x4, etc.

In the illustrated (x,y) coordinate system, a pixel is characterized by a width and height of 1. For example, the pixel 402 is defined in an area of the (x,y) coordinate system where x ranges from –0.5 to +0.5 and y ranges from –0.5 to +0.5 with the center of the pixel being an origin that has (0,0) coordinate. The offset pixels 404, 406, 408, and 410 are also characterized by the offset centers, which correspond to the pixel offset locations 414, 416, 418, and 420, respectively. Each of the pixel offset locations 414, 416, 418, and 420 of the offset pixels lies on or within the area defined by the pixel 402.

Optimum antialias solutions in this method ensure: 1) that image synthesis is done to subpixel resolutions of 1/N; and 2) that each subpixel axis define by the subpixel resolution of 1/N has one and only one offset pixel contribution, both within the pixel and in some cases shared with adjoining pixel locations.

Figures 5A through 5H illustrate exemplary pixel offset location patterns having four pixel offset locations that are defined in an (x,y) coordinate system for antialiasing a pixel. The pixel offset locations of these patterns maximize spatial diversity among the offset pixels, while providing a unique spatial position to each of the pixel offset locations in (x,y) coordinate.

Figure 5A shows pixel offset locations 502, 504, 506, and 508 having (x,y) coordinates (0.5,0), (0,-0.25), (-0.25,0.25), and (0.25,0.5), respectively. Figure 5B shows pixel offset locations 510, 512, 514, and 516 having (x,y) coordinates of (0.5,-0.25), (0,-0.5), (-0.25,0), and (0.25,0.25), respectively. Figure 5C depicts pixel offset locations 518, 520, 522, and 524 having (x,y) coordinates of (0.25,-0.25), (-0.25,0), (0,0.5), and (0.5,0.25), respectively. Figure 5D illustrates pixel offset locations 526, 528, 530, and 532 having (x,y) coordinates (0.5,0), (0.25,-0.5), (-0.25,-0.25), and (0,0.25), respectively. Figure 5E shows pixel offset locations 534, 536, 538, and 540 having (x,y) coordinates of (0.25,-0.25), (-0.25,-0.5), (-0.5,0), and (0,0.25), respectively. Figure 5F depicts pixel offset locations 542, 544, 546, and 548 having (x,y) coordinates of (0.25,0), (-0.25,-0.25), (-0.5,0.25), and (0,0.5), respectively. Figure 5G illustrates pixel offset locations 550, 552, 554, and 556 having (x,y) coordinates (0.25,0), (0,-0.5), (-0.5,-0.25), and (-0.25,0.25), respectively. Figure 5H shows pixel offset locations 558, 560, 562, and 564 having (x,y) coordinates of (0.25,0.25), (0,-0.25), (-0.5,0), and (-0.25,0.5), respectively.

Other pixel offset locations may also be used in accordance with some embodiments of the present invention. For example, Figure 6A illustrates a pixel offset location pattern 600 having eight pixel offset locations 602, 604, 606, 608, 610, 612, 614, and 616, which are characterized by (x,y) coordinates (3/8,-1/8), (1/8,-3/8), (-1/8,-1/4), (-3/8,0), (-1/4,1/4), (0,1/2), (1/4,3/8), and (1/2,1/8), respectively. Seven other similar pixel offset location patterns may be

generated by simply rotating the pixel offset locations 602, 604, 606, 608, 610, 612, 614, and 616 with respect to the x-axis, y-axis, 45 degree axis, and 135 degree axis.

Figure 6B shows another exemplary pixel offset location pattern 650. The pattern 650 has six pixel offset locations 652, 654, 656, 658, 660, and 662, which are characterized by (x,y) coordinates (1/2,0), (1/6,-1/3), (-1/6,-1/6), (-1/3,1/6), (0,1/2), and (1/3,1/3), respectively. In this case, seven other similar pixel offset location patterns may also be generated by simply rotating the pixel offset locations 652, 654, 656, 658, 660, and 662 with respect to the x-axis, y-axis, 45 degree axis, and 135 degree axis.

Figures 7A through 7D illustrate square and diamond pixel offset location patterns in accordance with certain embodiments of the present invention. These embodiments, unlike other examples given in this method, are suboptimum in that the subpixel axis have more than one offset pixel contribution, both within the pixel and in some cases shared with adjoining pixel locations. For example, Figure 7A shows an exemplary square pixel offset location pattern 700. The pixel offset location pattern 700 includes four pixel offset locations 702, 704, 706, and 708 having (x,y) coordinates (0.25,0.25), (0.25,-0.25), (-0.25,-0.25), and (-0.25,0.25), respectively. Figure 7B depicts a diamond pixel offset location pattern 720 having four pixel offset locations 722, 724, 726, and 728, which are characterized by (x,y) coordinates of (0.5,0), (0,-0.5), (-0.5,0), and (0,0.5), respectively. Figure 7C shows another exemplary square pixel offset location pattern 740. The square pixel offset location pattern 740 includes four pixel offset locations 742, 744, 746, and 748 having (x,y) coordinates of (0.5,0.5), (0.5,-0.5), (-0.5,-0.5), and (-0.5,0.5), respectively. Figure 7D illustrates another diamond pixel offset location pattern 760 having four pixel offset locations 762, 764, 766, and 768, which are characterized by (x,y) coordinates (0.25,0), (0,-0.25), (-0.25,0), and (0,0.25).

Figures 8A through 8D illustrate triangular pixel offset location patterns in accordance with some embodiments of the present invention. For example, Figure 8A shows a triangular pixel offset location pattern 800 having three pixel offset locations 802, 804, and 806, which are characterized by (x,y) coordinates of (1/3,0), (-1/3,-1/3), and (0,1/3), respectively. Figure 8B depicts a pixel offset location pattern 810, which has pixel offset locations 812, 814, and 816. The pixel offset locations 812, 814, and 816 are characterized by (x,y) coordinates (1/3,0), (0,-1/3), and (-1/3,1/3), respectively. Figure 8C illustrates another pixel offset location pattern 820, having pixel offset locations 822, 824, and 826, which are characterized by (x,y) coordinates of (1/3,-1/3), (-1/3,0), and (0,1/3). Figure 8D shows another triangular pixel offset pattern 830

having pixel offset locations 832, 834, and 836. The pixel offset locations 832, 834, and 836 have (x,y) coordinates (1/3,1/3), (0,-1/3), and (-1/3,0).

Figures 9A and 9B show pixel offset location patterns 900 and 950, respectively, each of which includes five pixel offset locations. Figure 9A illustrates the pixel offset location pattern 900 having pixel offset locations 902, 904, 906, 908, and 910. The pixel offset locations 902, 904, 906, 908, and 910 are defined by (x,y) coordinates (0,0), (0.4,-0.2), (-0.2,-0.4), (-0.4,0.2), and (0.2,0.4), respectively. Figure 9B depicts the pixel offset location pattern 950 having pixel offset locations 952, 954, 956, 958, and 960, which are characterized by (x,y) coordinates of (0.4,-0.2), (0,-0.4), (-0.4,0), (-0.2,0.4), and (0.2,0.2).

Figures 10A and 10B illustrate pixel offset location patterns 1000 and 1050, respectively, each of which includes seven pixel offset locations. Figure 10A shows an exemplary pixel offset location pattern 1000 having seven pixel offset locations. The pixel offset location pattern 1000 includes pixel offset locations 1002, 1004, 1006, 1008, 1010, 1012, and 1014, The pixel offset locations 1002, 1004, 1006, 1008, 1010, 1012, and 1014 are defined by (x,y) coordinates (0,0), (2/7,-2/7), (-1/7,-3/7), (-3/7,-1/7), (-2/7,2/7), (1/7,3/7), and (3/7,1/7), respectively. Figure 10B illustrates the pixel offset location pattern 1050 having pixel offset locations 1052, 1054, 1056, 1058, 1060, 1062, and 1064, which are positioned at (x,y) coordinates (3/7,0), (2/7,-2/7), (0,-3/7), (-2/7,-1/7), (-3/7,1/7), (-1/7,3/7), and (1/7, 2/7).

Figure 11 shows a flow chart of a method performed by the antialiasing unit 114 in accordance with one embodiment of the present invention. The method starts in operation 1102 and proceeds to operation 1104, where the antialiasing unit 114 generates a pixel clock PCLK, HCLK, and VSYNC signals. Then in operation 1106, the antialiasing unit 114 transmits PCLK, HSYNC, and VSYNC signals to each of the N graphics subsystems 112. In operation 1108, pixel data along with PCLK and a blanking signal from the graphics subsystems 112 are received.

The antialiasing unit 114, in operation 1110, selects one of the PCLK from the graphics subsystems 112 as a master pixel clock for use in processing pixel data. Optionally, multiple pixel clocks from graphics subsystems 112 can be used for some portion of the processing of pixel data, or the PCLK generated in the Antialiasing Unit 114 can be used to process pixel data. Then in operation 1112, the antialiasing unit 114 averages the pixel data (e.g., offset pixel data) received from the graphics subsystems 112 to generate antialiased pixel data in reference to the master pixel clock. In a preferred embodiment, the antialiasing unit 114 averages the pixel data using adders and registers that provide sufficient significant bits to generate larger pixel data

width. The antialiasing unit 114 then outputs the antialiased pixel data for display. Optionally in operation 1116, the antialiasing unit 114 may delay HSYNC and VSYNC signals relative input pixel data by the number of register states in the antialiasing unit 114. The method then terminates in operation 1118.

Figure 12 illustrates a more detailed schematic diagram of the antialiasing unit 114 for N=4 graphics subsystems in accordance with one embodiment of the present invention. The antialiasing unit 114 is arranged to use adders 1202, 1204, and 1206 to generate an output that is one bit wider than either of the input data for each adder. This technique allows the antialiasing unit 114 to perform loss-less averaging operations by addition without using complex division circuits.

The antialiasing unit 114 includes three adders 1202, 1204, and 1206 for adding pixel data (e.g., offset pixel data) from four graphics subsystems (GS) 202, 204, 206, and 208. Each of the adders is configured to provide a sum that is one significant bit wider than any of the input data width so that each add operation is effectively an averaging operation of two input values without the loss of pixel data resolution.

A plurality of m-bit registers 1218, 1220, 1222, and 1224 are provided at the input of each of the adders to buffer input pixel data from the graphics subsystems 202, 204, 206, and 208, respectively. The input pixel data from each of the graphics subsystems 202, 204, 206, and 208 are m bits per color component (R, G, or B) per pixel. In a preferred embodiment, m is 8. It should be noted that input pixel data includes R, G, and B components and thus includes three times the bits (e.g., 24 bits) of an individual component. A set of four of these R, G, and B color components are added in the antialiasing unit 114.

The registers 1218 and 1220 transmit the buffered m-bit input pixel data as inputs to the adder 1202 while the registers 1222 and 1224 feed the buffered m-bit input pixel data as inputs to the adder 1204. Each of the adders 1202 and 1204 adds the respective m-bit input pixel data and outputs an m+1 bit sum. The (m+1) bit sum from the adder 1202 is buffered in an (m+1)-bit register 1226 while the (m+1) bit sum from the adder 1204 is buffered in an (m+1)-bit register 1228. The (m+1) bit sum from the register 1226 is then input into the adder 1206 as an input. Similarly, the (m+1) bit sum from the register 1228 is also input into the adder 1206 as another input. The adder 1206 adds the (m+1) bit input pixel data and generates an (m+2) bit output sum. The (m+2) bit output sum is then buffered into an (m+2) bit register 1230. The register 1230 then transmits the higher resolution (m+2) bit sum, which represents an average of all the input pixel data, to the DAC 116. In a preferred embodiment, the DAC 116 receives 10 bit data

for each color component (R, G, and B), which began as an 8-bit component. Hence, the averaging in the antialiasing unit 114 is performed without loss of precision or resolution.

The antialiasing unit 114 also includes a clock generator 1208 and a sync generator 120 for synchronizing pixel data processing. The clock generator 1208 generates pixel clock PCLK while the sync generator produces HSYNC, and VSYNC signals. The pixel clock PCLK is transmitted to each of the graphics subsystems 202, 204, 206, and 208 for use as a reference clock. The sync generator 1210 provides the HSYNC and VSYNC signals to the graphics subsystems 202, 204, 206, and 208 for synchronizing pixel data processing. In addition, the sync generator 1210 transmits the HSYNC and VSYNC signals to the display device 118 for synchronizing display of pixel data signals. Optionally, the antialiasing unit 114 can transmit HSYNC and/or VSYNC signals from one of the graphics subsystems 112, or from an external source, to drive the display device 118.

To provide synchronized blanking signal, the antialiasing unit 114 receives the blanking signal from one or more of the graphics subsystems 202, 204, 206, and 208 and delays the blanking signal by the same register states provided in the datapath for the addition operations. For example, since the addition datapath includes three layers of registers, the antialiasing unit 114 includes three registers 1212, 1214, and 1216 to match the delay of the pixel data through the add operation. This arrangement allows the blanking signal to be transmitted to the DAC 116 at a same clock cycle as the antialiased pixel data output of the register 1230.

The DAC 208 is coupled to the antialiasing unit 206 to receive the antialiased pixel data and the blanking signal for converting the pixel data into analog RGB signals in a synchronous manner. The DAC 208 then transmits the analog RGB signals to the display device 118 for display, which receives VSYNC and HSYNC from the antialiasing unit 114 to synchronize the display image.

The present invention thus generates antialiased pixel data at substantially real time without the delay and reduction in video bandwidth associated with conventional accumulation and averaging of display images over time. By associating a set of graphics subsystems with a set of pixel offset locations to generate offset pixels, the present invention also provides scalability and flexibility in adapting to pixel resolutions. Further, the present invention also provides an antialiasing unit that generates antialiased pixel data by averaging the input offset pixel data using a set of adders adapted to add m-bit input R, G, and B components of the offset pixel data to generate (m+1) bit output R, G, B components. For N offset pixel data inputs of m-

bit each, the antialiasing unit may generate ($m+\log_2 N$) bit antialiased pixel data. This arrangement allows loss-less antialiasing even at a substantially high resolution.

Methods and systems have thus been described for efficiently antialiasing pixel data to generate quality medium- to high-resolution images without employing a costly computational efforts or large amounts of specialized hardware. The system and method reduce or eliminate jagged and crawling effects of lines and silhouette edges, scintillation effects on small objects and very thin lines, and low-frequency aliases and Moire interference patterns in textures. In addition, the present invention intensifies thin off-axis lines to offset the effect of longer distances between points on off-axis lines by filtering. Additionally the method and system allow such antialiasing without substantial delay or video bandwidth limitations so as to meet real time requirements of performance critical applications.

While this invention has been described in terms of several preferred embodiments, there are alterations, permutations, and equivalents which fall within the scope of this invention. It should also be noted that there are alternative ways of implementing both the methods and devices of the present invention. It is therefore intended that the following appended claims be interpreted as including all such alterations, permutations, and equivalents as fall within the true spirit and scope of the present invention.

What is claimed is:

CLAIMS

1. In a computer system, a method for antialiasing a pixel, the pixel being defined by pixel data, the method comprising:

providing a set of pixel offset locations for the pixel, each of the pixel offset locations being provided relative to a center of the pixel, each of the pixel offset locations being unique from every other pixel offset locations;

generating a set of offset pixels at the pixel offset locations associated with the pixel, one offset pixel per pixel offset location, each of the offset pixels being defined by offset pixel data; and

filtering the set of offset pixels at the pixel offset locations to generate antialiased pixel data for the pixel.

2. The method as recited in claim 1, wherein the operation of providing the unique pixel offset locations comprises:

providing a set of graphics subsystems arranged to the set of offset pixels; and

associating the set of graphics subsystems with the set of offset pixels associated with the pixel offset locations, one graphics subsystem being associated with one unique pixel offset location.

3. The method as recited in claim 2, wherein the set of graphics subsystems generates the set of offset pixels at the associated pixel offset locations, one graphics subsystem generating one offset pixel.

4. The method as recited in claim 1, wherein the filtering comprises:

averaging the set of offset pixels associated with the set of pixel offset locations.

5. The method as recited in claim 4, wherein the operation of averaging comprises adding the set of offset pixel data.

6. The method as recited in claim 1, further comprising:

converting the antialiased pixel data into an analog signal; and

displaying the analog signal corresponding to the antialiased pixel data.

7. The method as recited in claim 1, wherein the set of pixel offset locations contains N pixel offset locations and the set of offset pixels contains N offset pixels.

8. The method as recited in claim 2, wherein the set of pixel offset locations contains N pixel offset locations and the set of graphics subsystems contains N subsystems.

9. The method as recited in claim 7, wherein the antialiased pixel is generated from the set of offset pixels.

10. The method as recited in claim 9, wherein the pixel defines an area that lies in an (x,y) coordinate system where x ranges from –0.5 to +0.5 and y ranges from –0.5 to +0.5 with the center of the pixel being an origin that has (0,0) coordinate.

11. The method as recited in claim 10, wherein each of the offset pixels has an offset center and an offset pixel area, the offset center of each of the offset pixels corresponding to the associated pixel offset location, wherein each of the pixel offset locations of the offset pixels lies on or within the area defined by the pixel.

12. The method as recited in claim 11, wherein each of the offset pixel area of the offset pixels is equal to the area of the pixel.

13. The method as recited in claim 11, wherein the pixel offset locations are adapted to maximize spatial diversity among the offset pixels.

14. The method as recited in claim 11, wherein N is 4.

15. The method as recited in claim 14, wherein the (x,y) coordinates of the pixel offset locations of the four offset pixels include (0.5,0), (0,-0.25), (-0.25,0.25), and (0.25,0.5).

16. The method as recited in claim 14, wherein the (x,y) coordinates of the pixel offset locations of the four offset pixels include (0.5,-0.25), (0,-0.5), (-0.25,0), and (0.25,0.25).

17. The method as recited in claim 14, wherein the (x,y) coordinates of the offset pixel locations of the four offset pixels include (0.25,-0.25), (-0.25,-0.5), (-0.5,0), and (0,0.25).

18. The method as recited in claim 14, wherein the (x,y) coordinates of the pixel offset locations of the four offset pixels include (0.25,0), (-0.25,-0.25), (-0.5,0.25), and (0,0.5).

19. The method as recited in claim 14, wherein the (x,y) coordinates of the pixel offset locations of the four offset pixels include (0.25,-0.25), (-0.25,0), (0,0.5), and (0.5,0.25).

20. The method as recited in claim 14, wherein the (x,y) coordinates of the pixel offset locations of the four offset pixels include (0.5,0), (0.25,-0.5), (-0.25,-0.25), and (0,0.25).

21. The method as recited in claim 14, wherein the (x,y) coordinates of the pixel offset locations of the four offset pixels include (0.25,0), (0,-0.5), (-0.5,-0.25), and (-0.25,0.25).

22. The method as recited in claim 14, wherein the (x,y) coordinates of the pixel offset locations of the four offset pixels include (0.25,0.25), (0,-0.25), (-0.5,0), and (-0.25,0.5).

23. The method as recited in claim 14, wherein the (x,y) coordinates of the pixel offset locations of the four offset pixels include (0.25,0.25), (0.25,-0.25), (-0.25,-0.25), and (-0.25,0.25).

24. The method as recited in claim 14, wherein the (x,y) coordinates of the pixel offset locations of the four offset pixels include (0.5,0), (0,-0.5), (-0.5,0), and (0,0.5).

25. The method as recited in claim 14, wherein the (x,y) coordinates of the pixel offset locations of the four offset pixels include (0.5,0.5), (0.5,-0.5), (-0.5,-0.5), and (-0.5,0.5).

26. The method as recited in claim 14, wherein the (x,y) coordinates of the pixel offset locations of the four offset pixels include (0.25,0), (0,-0.25), (-0.25,0), and (0,0.25).

27. The method as recited in claim 11, wherein N is 8.

28. The method as recited in claim 27, wherein the (x,y) coordinates of the pixel offset locations of the eight offset pixels include (3/8,-1/8), (1/8,-3/8), (-1/8,-1/4), (-3/8,0), (-1/4,1/4), (0,1/2), (1/4,3/8), and (1/2,1/8).

29. The method as recited to claim 27, wherein the eight pixel offset locations are rotated about an x-axis.

30. The method as recited to claim 27, wherein the eight pixel offset locations are rotated about a y-axis.

31. The method as recited to claim 27, wherein the eight pixel offset locations are rotated about a 45 degree straight line through (0,0).

32. The method as recited to claim 27, wherein the eight pixel offset locations are rotated about a 135 degree straight line through (0,0).

33. The method as recited to claim 27, wherein the rotated eight pixel offset locations are rotated about an x-axis.

34. The method as recited to claim 27, wherein the rotated eight pixel offset locations are rotated about a y-axis.

35. The method as recited to claim 27, wherein the rotated eight pixel offset locations are rotated about a 135 degree straight line through (0,0).

36. The method as recited in claim 11, wherein N is 6.

37. The method as recited in claim 36, wherein the (x,y) coordinates of the pixel offset locations of the six offset pixels include (1/2,0), (1/6,-1/3), (-1/6,-1/6), (-1/3,1/6), (0,1/2), and (1/3,1/3).

38. The method as recited to claim 36, wherein the six pixel offset locations are rotated about an x-axis.

39. The method as recited to claim 36, wherein the six pixel offset locations are rotated about a y-axis.

40. The method as recited to claim 36, wherein the six pixel offset locations are rotated about a 45 degree straight line through (0,0).

41. The method as recited to claim 36, wherein the six pixel offset locations are rotated about a 135 degree straight line through (0,0).

42. The method as recited to claim 36, wherein the rotated six pixel offset locations are rotated about an x-axis.

43. The method as recited to claim 36, wherein the rotated six pixel offset locations are rotated about a y-axis.

44. The method as recited to claim 36, wherein the rotated six (pixel offset locations are rotated about a 135 degree straight line through (0,0).

45. The method as recited in claim 11, wherein N is 3.

46. The method as recited in claim 45, wherein the (x,y) coordinates of the pixel offset locations of the offset pixels include (1/3,0), (-1/3,-1/3), and (0,1/3).

47. The method as recited in claim 46, wherein the (x,y) coordinates of the pixel offset locations of the offset pixels include (1/3,0), (0,-1/3), and (-1/3,1/3).

48. The method as recited in claim 46, wherein the (x,y) coordinates of the pixel offset locations of the offset pixels include (1/3,-1/3), (-1/3,0), and (0,1/3).

49. The method as recited in claim 46, wherein the (x,y) coordinates of the pixel offset locations of the offset pixels include (1/3,1/3), (0,-1/3), and (-1/3,0).

50. The method as recited in claim 11, wherein N is 5.

51. The method as recited in claim 50, wherein the (x,y) coordinates of the pixel offset locations of the offset pixels include (0,0), (0.4,-0.2), (-0.2,-0.4), (-0.4,0.2), and (0.2,0.4).

52. The method as recited in claim 50, wherein the (x,y) coordinates of the pixel offset locations of the offset pixels include (0.4,-0.2), (0,-0.4), (-0.4,0), (-0.2,0.4), and (0.2,0.2).

53. The method as recited in claim 11, wherein N is 7.

54. The method as recited in claim 53, wherein the (x,y) coordinates of the pixel offset locations of the offset pixels include (0,0), (2/7,-2/7), (-1/7,-3/7), (-3/7,-1/7), (-2/7,2/7), (1/7,3/7), and (3/7,1/7).

55. The method as recited in claim 53, wherein the (x,y) coordinates of the pixel offset locations of the of the offset pixels include (3/7,0), (2/7,-2/7), (0,-3/7), (-2/7,-1/7), (-3/7,1/7), (-1/7,3/7), and (1/7, 2/7).

56. A computer system for antialiasing a pixel defined by pixel data, comprising:

a bus;

a processor coupled to the bus;

a memory coupled to the bus;

a storage device coupled to the bus;

a set of graphics subsystems coupled to the bus and being associated with a set of pixel offset locations for the pixel, each of the pixel offset locations being provided relative to a center of the pixel, each of the pixel offset locations being unique from every other pixel offset locations, the set of graphics subsystems being arranged to generate a set of offset pixels at the pixel offset locations associated with the pixel, one offset pixel per pixel offset location, each of the offset pixels being defined by offset pixel data; and

an antialiasing unit coupled to receive the set of offset pixel data to filter the set of offset pixels at the pixel offset locations to generate antialiased pixel data for the pixel.

57. The system as recited in claim 56, wherein the set of graphics subsystems are associated with the set of offset pixels associated with the pixel offset locations, one graphics subsystem being associated with one spatially unique pixel offset location.

58. The system as recited in claim 57, wherein the set of graphics subsystems generates the set of offset pixels at the associated pixel offset locations, one graphics subsystem generating one offset pixel.

59. The system as recited in claim 56, wherein the antialiasing unit filters the set of offset pixels by averaging the set of offset pixels associated with the set of pixel offset locations.

60. The system as recited in claim 59, wherein the antialiasing unit averages the set of offset pixels by adding the set of offset pixel data.

61. The system as recited in claim 56, further comprising:

a DAC coupled to receive the antialiased pixel data and being adapted to convert the antialiased pixel data into an analog signal; and

a display device coupled to receive the analog signal for display.

62. The system as recited in claim 56, wherein the set of pixel offset locations contains N pixel offset locations and the set of offset pixels contains N offset pixels.

63. The system as recited in claim 57, wherein the set of pixel offset locations contains N pixel offset locations and the set of graphics subsystems contains N subsystems.

64. The system as recited in claim 62, wherein the antialiased pixel is generated from the set of offset pixels.

65. The system as recited in claim 64, wherein the pixel defines an area that lies in an (x,y) coordinate system where x ranges from –0.5 to +0.5 and y ranges from –0.5 to +0.5 with the center of the pixel being an origin that has (0,0) coordinate.

66. The system as recited in claim 65, wherein each of the offset pixels has an offset center and an offset pixel area, the offset center of each of the offset pixels corresponding to the associated pixel offset location, wherein each of the pixel offset locations of the offset pixels lies on or within the area defined by the pixel.

67. The system as recited in claim 66, wherein each of the offset pixel area of the offset pixels is equal to the area of the pixel.

68. The system as recited in claim 66, wherein the pixel offset locations are adapted to maximize spatial diversity among the offset pixels.

69. The system as recited in claim 66, wherein N is 4.

70. The system as recited in claim 65, wherein the (x,y) coordinates of the pixel offset locations include (0.5,0), (0,-0.25), (-0.25,0.25), and (0.25,0.5).

71. The system as recited in claim 65, wherein the (x,y) coordinates of the pixel offset locations include (0.5,-0.25), (0,-0.5), (-0.25,0), and (0.25,0.25).

72. The system as recited in claim 65, wherein the (x,y) coordinates of the offset pixel locations include (0.25,-0.25), (-0.25,-0.5), (-0.5,0), and (0,0.25).

73. The system as recited in claim 65, wherein the (x,y) coordinates of the pixel offset locations include (0.25,0), (-0.25,-0.25), (-0.5,0.25), and (0,0.5).

74. The system as recited in claim 65, wherein the (x,y) coordinates of the pixel offset locations include (0.25,-0.25), (-0.25,0), (0,0.5), and (0.5,0.25).

75. The system as recited in claim 65, wherein the (x,y) coordinates of the pixel offset locations include (0.5,0), (0.25,-0.5), (-0.25,-0.25), and (0,0.25).

76. The system as recited in claim 65, wherein the (x,y) coordinates of the pixel offset locations include (0.25,0), (0,-0.5), (-0.5,-0.25), and (-0.25,0.25).

77. The system as recited in claim 65, wherein the (x,y) coordinates of the pixel offset locations include (0.25,0.25), (0,-0.25), (-0.5,0), and (-0.25,0.5).

78. The system as recited in claim 56, wherein the offset pixel data includes m-bit R, G, and B components, wherein the antialiasing unit comprises:

a set of adders adapted to receive and add the m-bit R, G, and B components of the offset pixel data to generate (m+1) bit R, G, B component.

79. The system as recited in claim 78, wherein the set of graphics subsystems includes N graphics subsystems, wherein the antialiasing unit is arranged to generate $(m+\log_2 N)$ bit R, G, and B component as the antialiased pixel data.

80. A system for antialiasing a pixel defined by pixel data, comprising:

means for generating a set of offset pixels at a set of pixel offset locations, one offset pixel per pixel offset location, each of the offset pixels being defined by offset pixel data, the generating means being associated with the set of pixel offset locations for the pixel, each of the

pixel offset locations being provided relative to a center of the pixel, each of the pixel offset locations being unique from every other pixel offset locations; and

antialiasing means for filtering the set of offset pixels at the pixel offset locations to generate antialiased pixel data for the pixel.

81. The system as recited in claim 80, wherein the antialiasing means filters the set of offset pixels by averaging the set of offset pixels associated with the set of pixel offset locations.

82. The system as recited in claim 81, wherein the antialiasing means averages the set of offset pixels by adding the set of offset pixel data.

83. The system as recited in claim 80, further comprising:

means for converting the antialiased pixel data into an analog signal; and

means for displaying the analog signal.

84. The system as recited in claim 80, wherein the set of pixel offset locations contains N pixel offset locations and the set of offset pixels contains N offset pixels.

85. The system as recited in claim 80, wherein the antialiased pixel is generated from the set of offset pixels.

86. The system as recited in claim 84, wherein the pixel defines an area that lies in an (x,y) coordinate system where x ranges from –0.5 to +0.5 and y ranges from –0.5 to +0.5 with the center of the pixel being an origin that has (0,0) coordinate.

87. The system as recited in claim 86, wherein each of the offset pixels has an offset center and an offset pixel area, the offset center of each of the offset pixels corresponding to the associated pixel offset location, wherein each of the pixel offset locations of the offset pixels lies on or within the area defined by the pixel.

88. The system as recited in claim 87, wherein each of the offset pixel area of the offset pixels is equal to the area of the pixel.

89. The system as recited in claim 80, wherein the pixel offset locations are adapted to maximize spatial diversity among the offset pixels.

90. The system as recited in claim 84, wherein N is 4.

91. The system as recited in claim 86, wherein the (x,y) coordinates of the pixel offset locations include (0.5,0), (0,-0.25), (-0.25,0.25), and (0.25,0.5).

92. The system as recited in claim 86, wherein the (x,y) coordinates of the pixel offset locations include (0.5,-0.25), (0,-0.5), (-0.25,0), and (0.25,0.25).

93. The system as recited in claim 86, wherein the (x,y) coordinates of the offset pixel locations include (0.25,-0.25), (-0.25,-0.5), (-0.5,0), and (0,0.25).

94. The system as recited in claim 86, wherein the (x,y) coordinates of the pixel offset locations include (0.25,0), (-0.25,-0.25), (-0.5,0.25), and (0,0.5).

95. The system as recited in claim 86, wherein the (x,y) coordinates of the pixel offset locations include (0.25,-0.25), (-0.25,0), (0,0.5), and (0.5,0.25).

96. The system as recited in claim 86, wherein the (x,y) coordinates of the pixel offset locations include (0.5,0), (0.25,-0.5), (-0.25,-0.25), and (0,0.25).

97. The system as recited in claim 86, wherein the (x,y) coordinates of the pixel offset locations include (0.25,0), (0,-0.5), (-0.5,-0.25), and (-0.25,0.25).

98. The system as recited in claim 86, wherein the (x,y) coordinates of the pixel offset locations include (0.25,0.25), (0,-0.25), (-0.5,0), and (-0.25,0.5).

99. The system as recited in claim 80, wherein the offset pixel data includes m-bit R, G, and B components, wherein the antialiasing means comprises:

a set of adders adapted to receive and add the m-bit R, G, and B components of the offset pixel data to generate (m+1) bit R, G, B component.

100. The system as recited in claim 80, wherein the antialiasing means is arranged to receive N m-bit offset pixel data as inputs, wherein the antialiasing means is adapted to generate $(m+\log_2 N)$ bit antialiased pixel data.

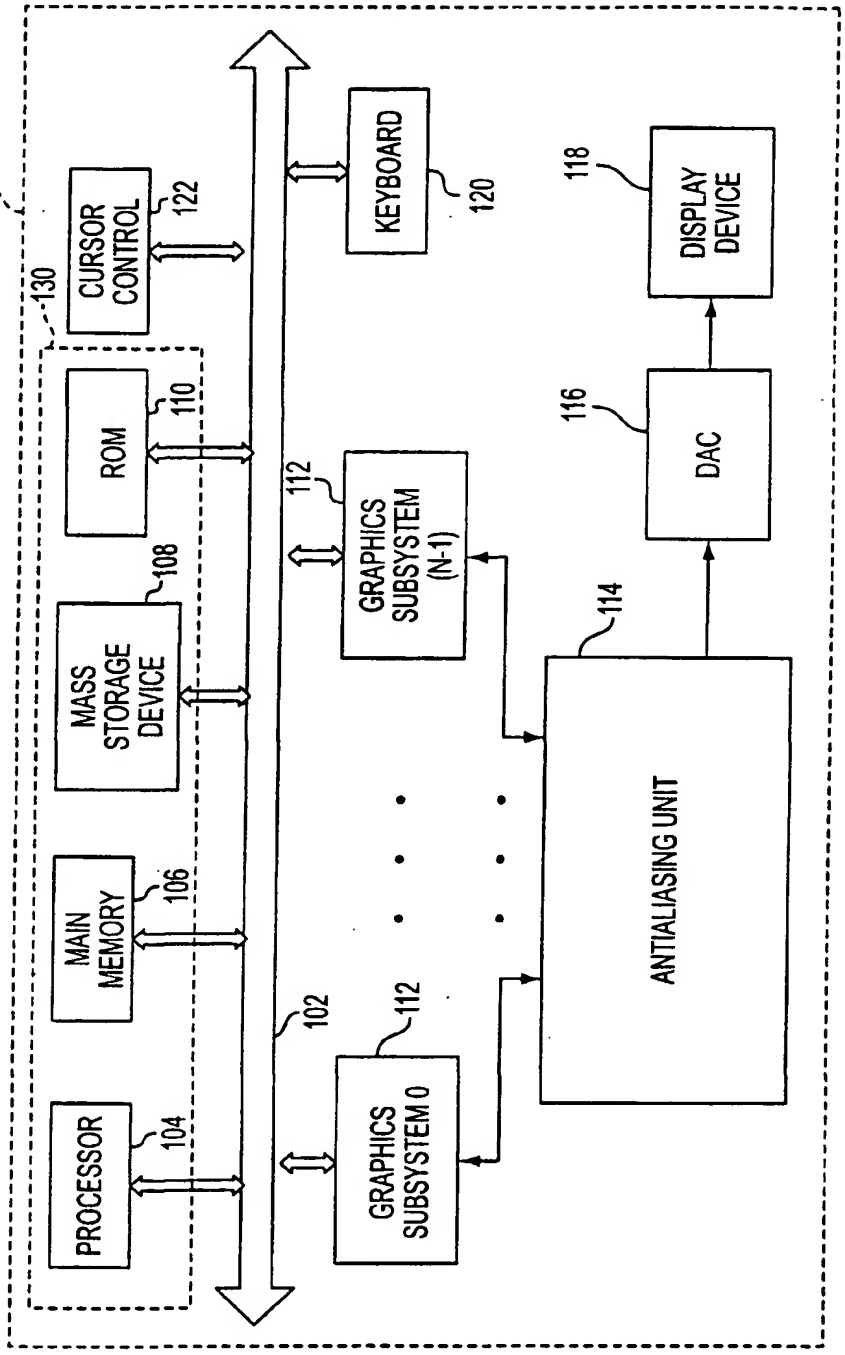

100
130
CURSOR CONTROL
122
KEYBOARD
120
118
DISPLAY DEVICE
ROM
110
116
DAC
112
GRAPHICS SUBSYSTEM (N-1)
108
MASS STORAGE DEVICE
114
ANTIALIASING UNIT
MAIN MEMORY
106
102
112
GRAPHICS SUBSYSTEM 0
PROCESSOR
104

FIG 1A

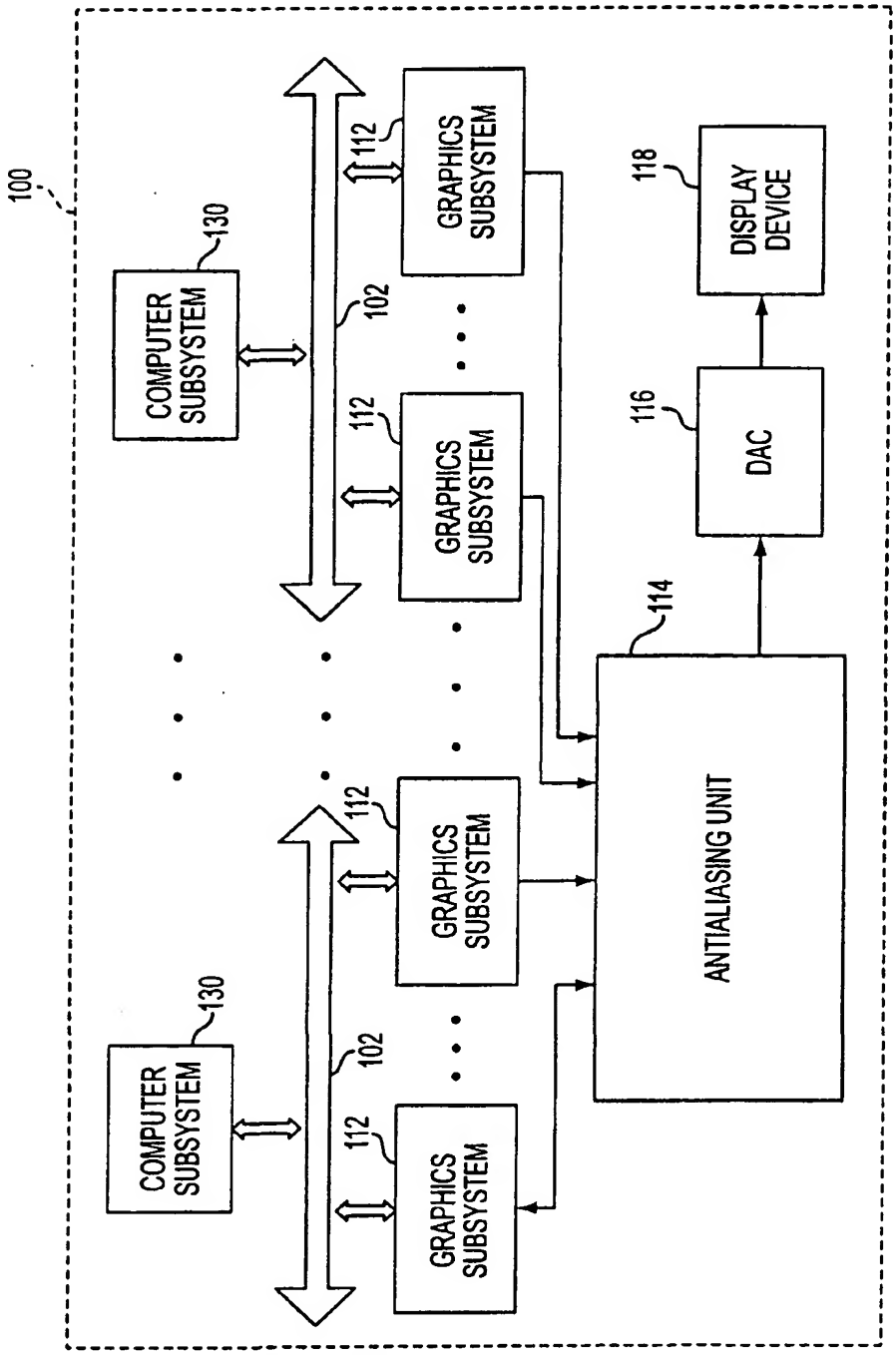

100
COMPUTER SUBSYSTEM
130
COMPUTER SUBSYSTEM
130
102
102
112
GRAPHICS SUBSYSTEM
112
GRAPHICS SUBSYSTEM
112
GRAPHICS SUBSYSTEM
112
GRAPHICS SUBSYSTEM
ANTIALIASING UNIT
114
DAC
116
DISPLAY DEVICE
118

FIG 1B

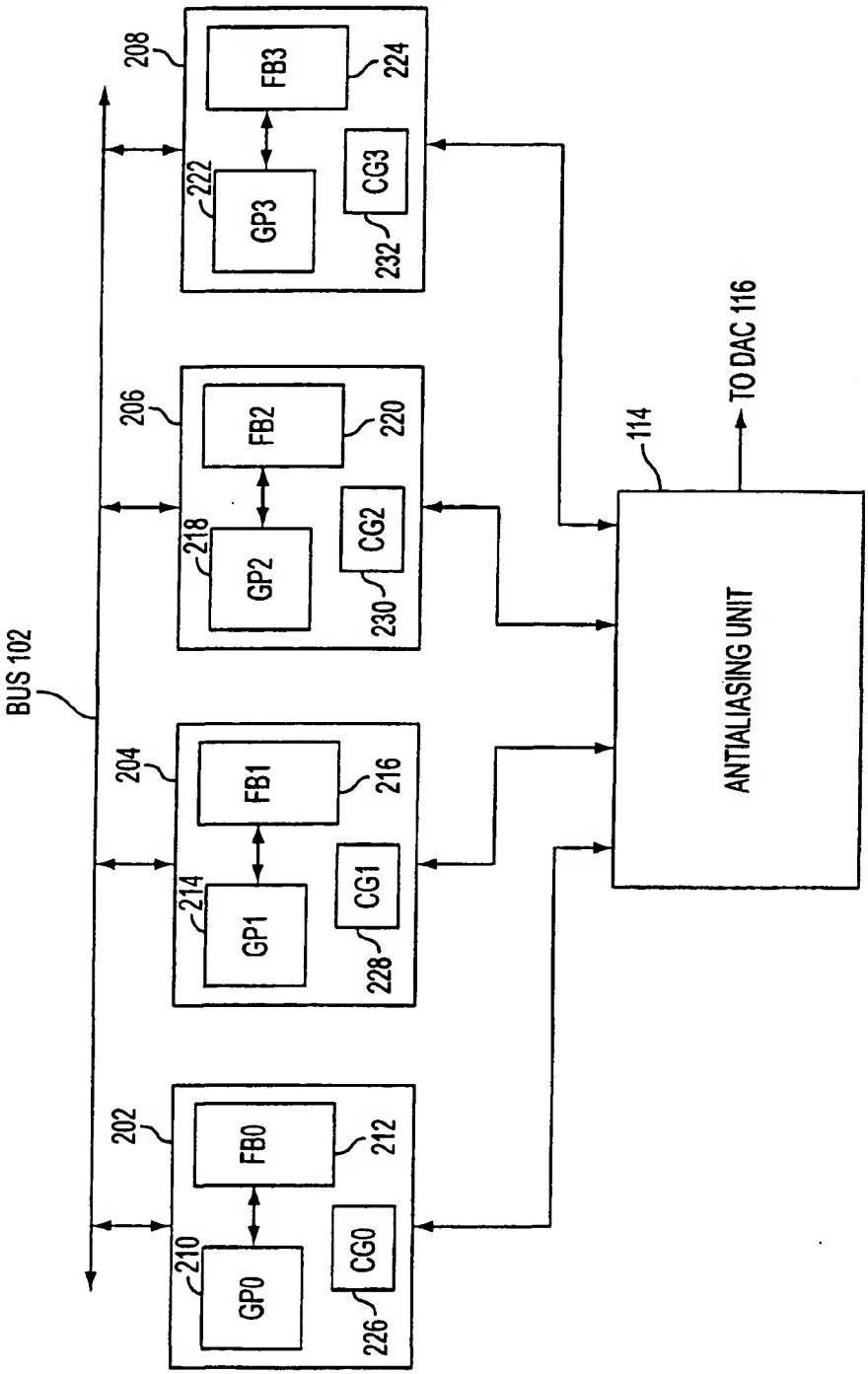

BUS 102
208
FB3
224
222
GP3
CG3
232
206
FB2
220
218
GP2
CG2
230
204
FB1
216
214
GP1
CG1
228
202
FB0
212
210
GP0
CG0
226
114
TO DAC 116
ANTIALIASING UNIT

FIG. 2

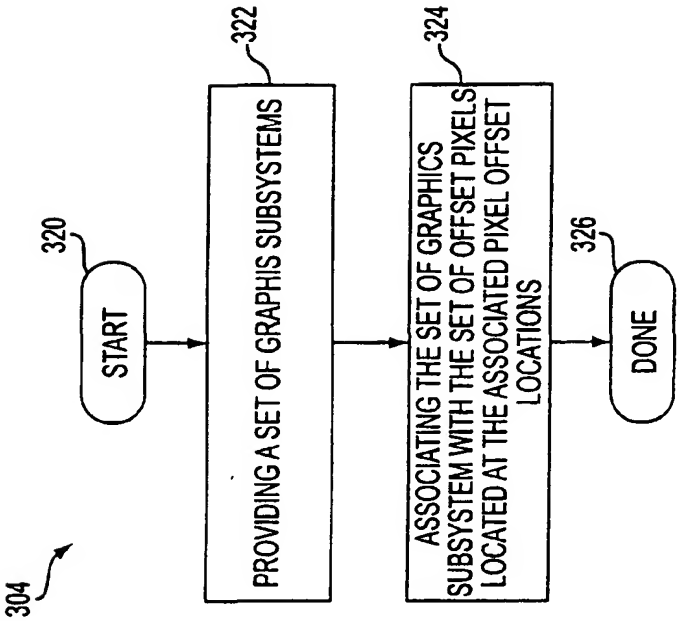

304
320
START
322
PROVIDING A SET OF GRAPHIS SUBSYSTEMS
324
ASSOCIATING THE SET OF GRAPHICS SUBSYSTEM WITH THE SET OF OFFSET PIXELS LOCATED AT THE ASSOCIATED PIXEL OFFSET LOCATIONS
326
DONE

FIG. 3B

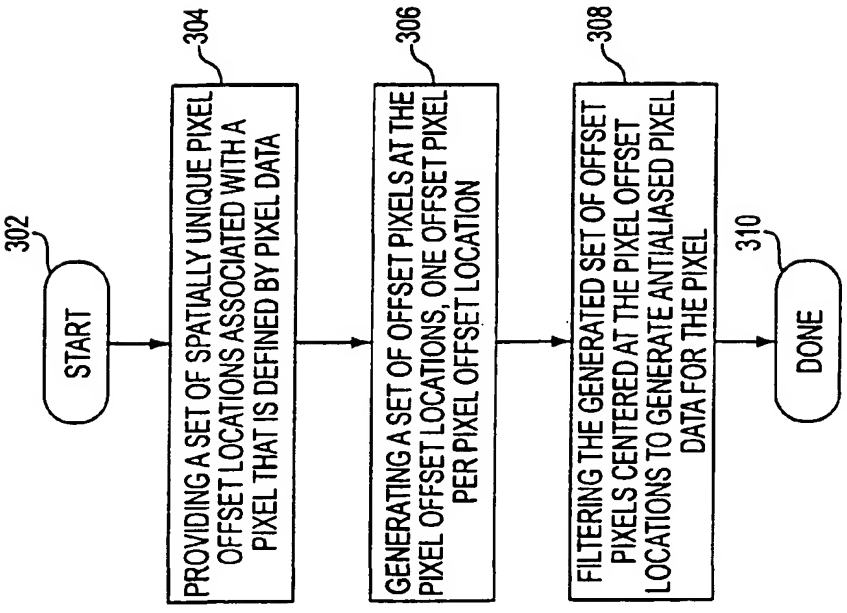

302
START
304
PROVIDING A SET OF SPATIALLY UNIQUE PIXEL OFFSET LOCATIONS ASSOCIATED WITH A PIXEL THAT IS DEFINED BY PIXEL DATA
306
GENERATING A SET OF OFFSET PIXELS AT THE PIXEL OFFSET LOCATIONS, ONE OFFSET PIXEL PER PIXEL OFFSET LOCATION
308
FILTERING THE GENERATED SET OF OFFSET PIXELS CENTERED AT THE PIXEL OFFSET LOCATIONS TO GENERATE ANTIALIASED PIXEL DATA FOR THE PIXEL
310
DONE

FIG. 3A

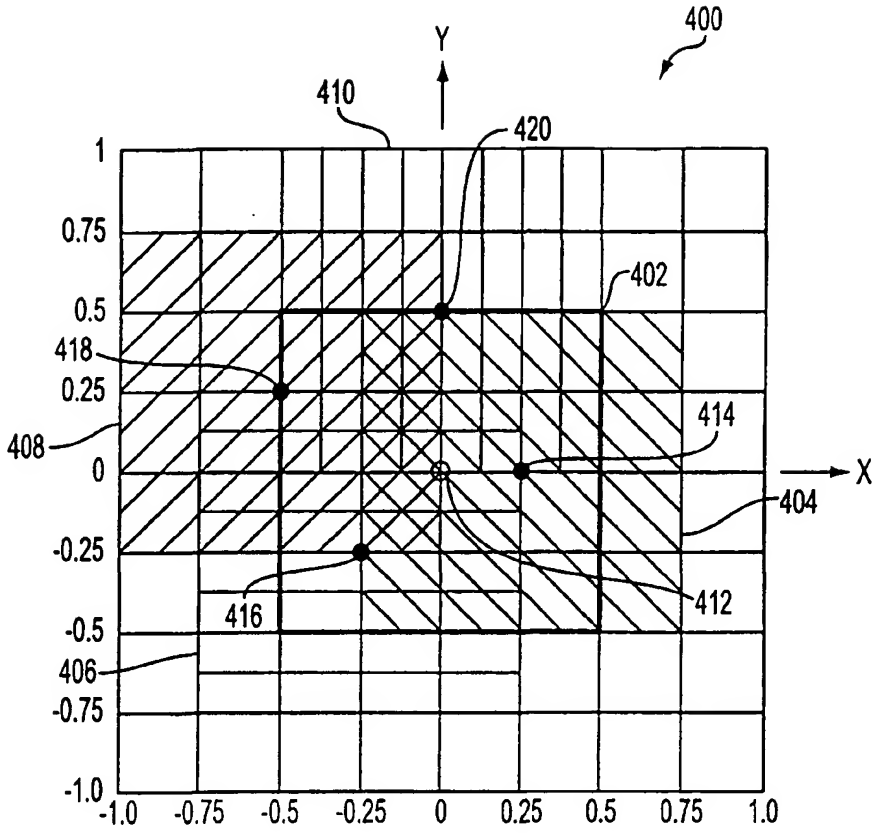

400
Y
410
420
1
0.75
402
0.5
418
0.25
414
408
0
X
404
-0.25
412
416
-0.5
406
-0.75
-1.0
-1.0 -0.75 -0.5 -0.25 0 0.25 0.5 0.75 1.0

FIG. 4

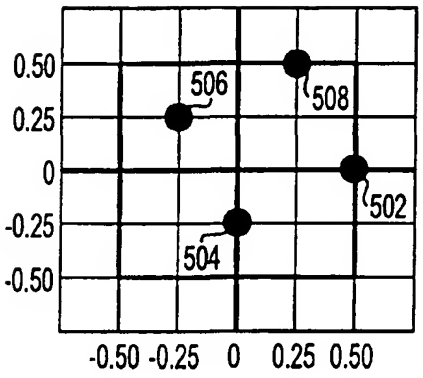


FIG. 5A

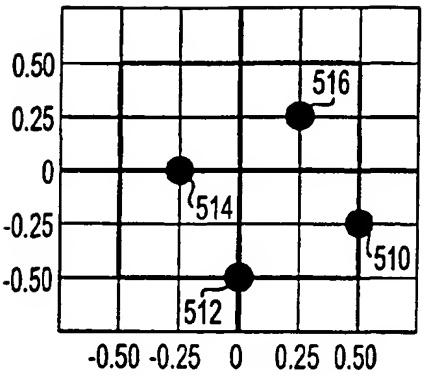


FIG. 5B

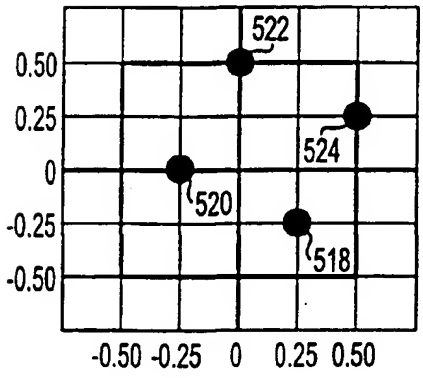


FIG. 5C

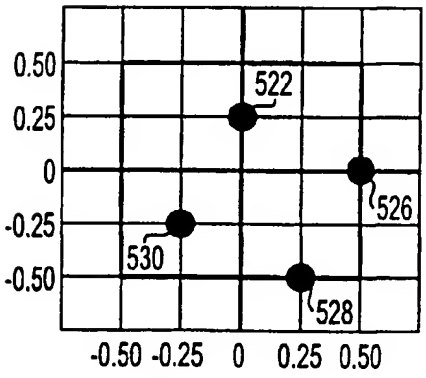


FIG. 5D

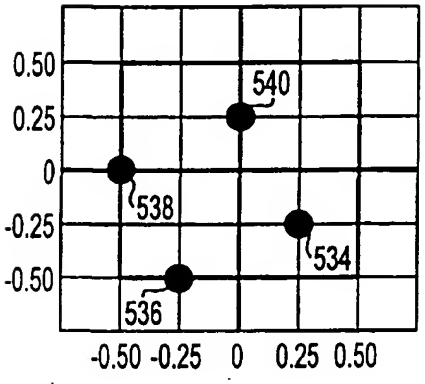


FIG. 5E

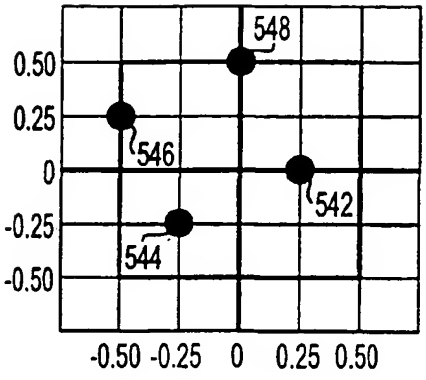


FIG. 5F

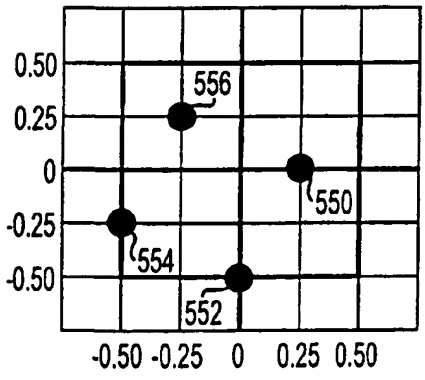


FIG. 5G

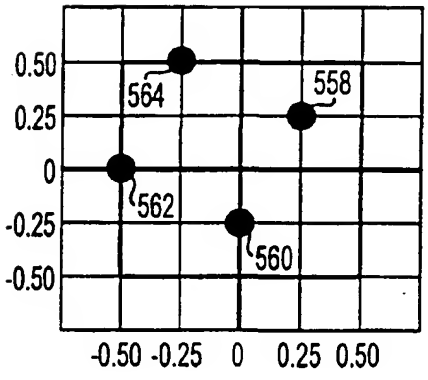


FIG. 5H

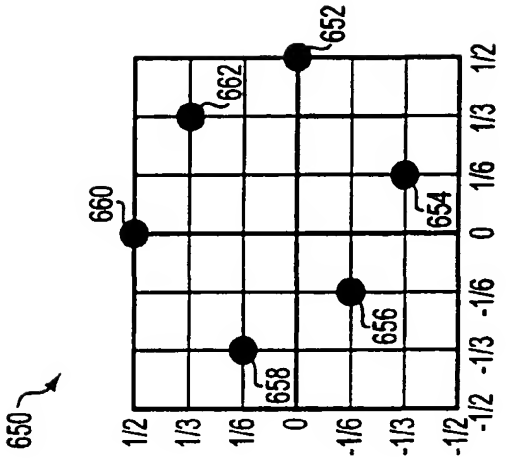

650
660
662
652
654
656
658
1/2
1/3
1/6
0
-1/6
-1/3
-1/2

FIG. 6B

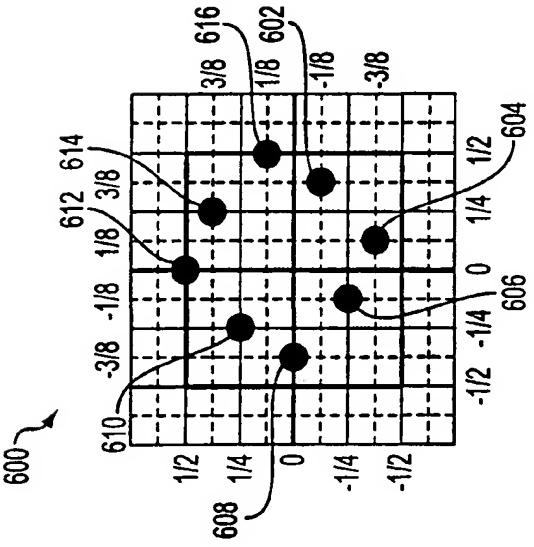

600
610
612
614
616
602
604
606
608
3/8
1/8
-1/8
-3/8
1/2
1/4
0
-1/4
-1/2

FIG. 6A

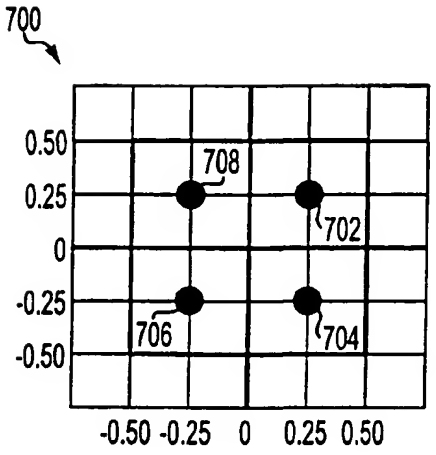


FIG. 7A

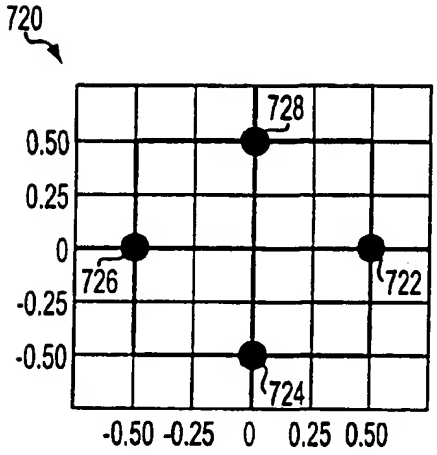


FIG. 7B

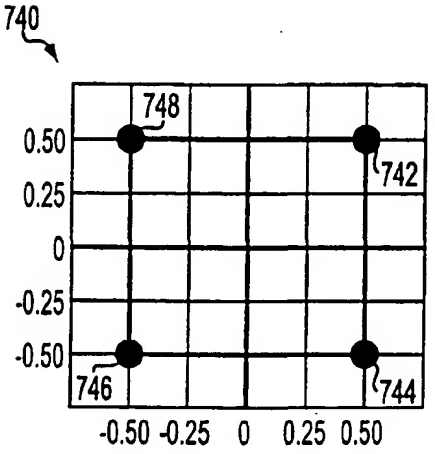


FIG. 7C

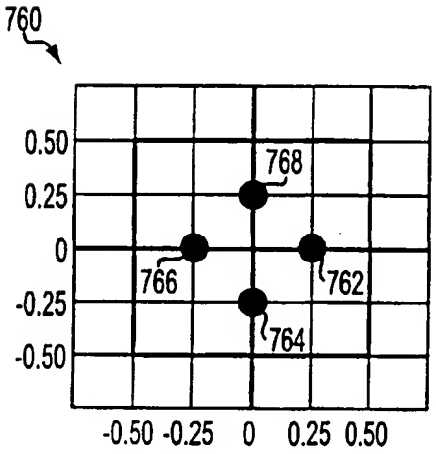


FIG. 7D

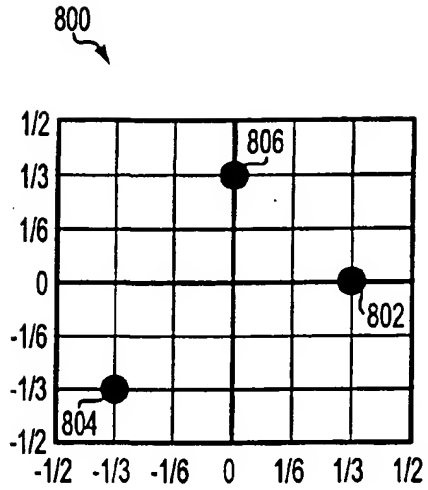


FIG. 8A

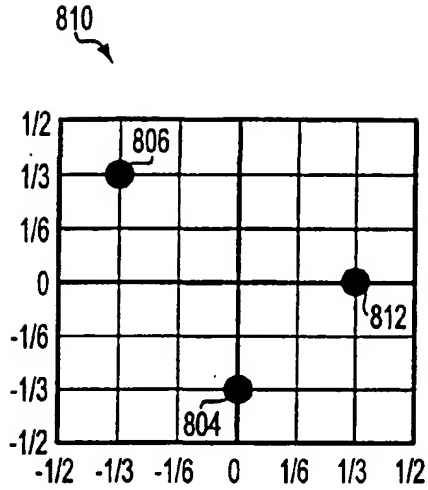


FIG. 8B

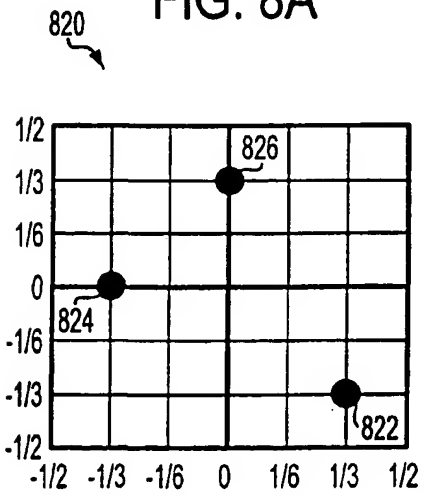


FIG. 8C

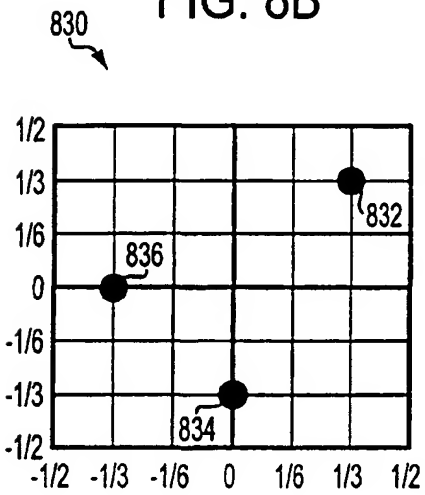


FIG. 8D

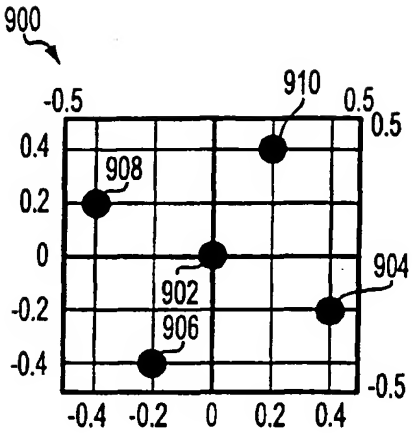


FIG. 9A

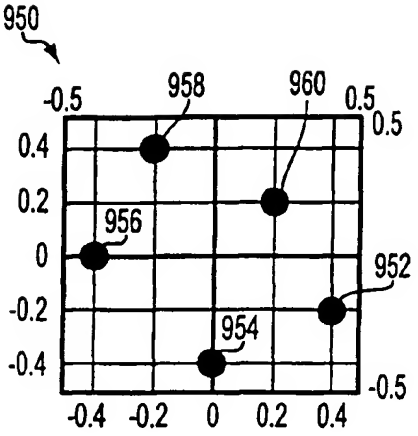


FIG. 9B

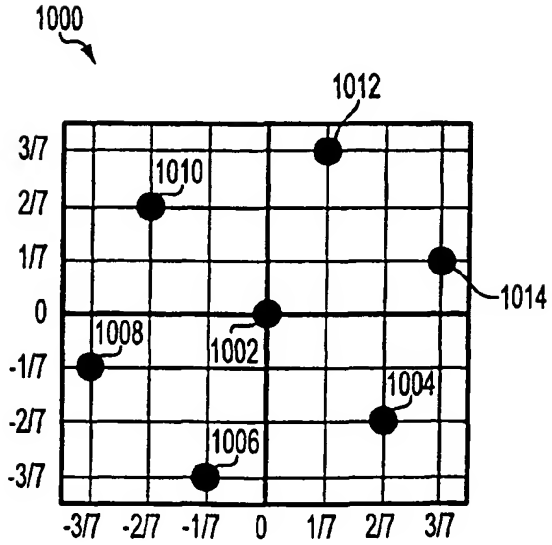


FIG. 10A

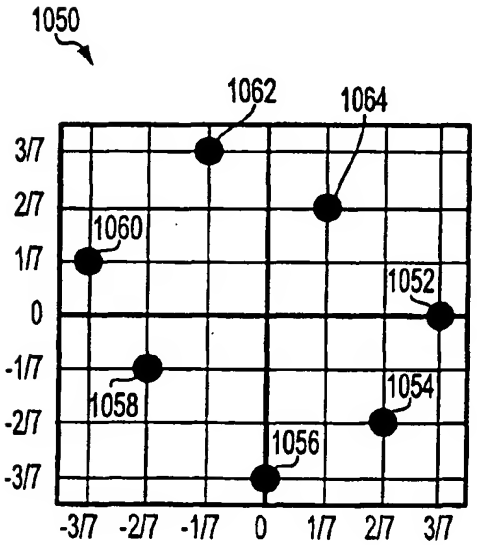


FIG. 10B

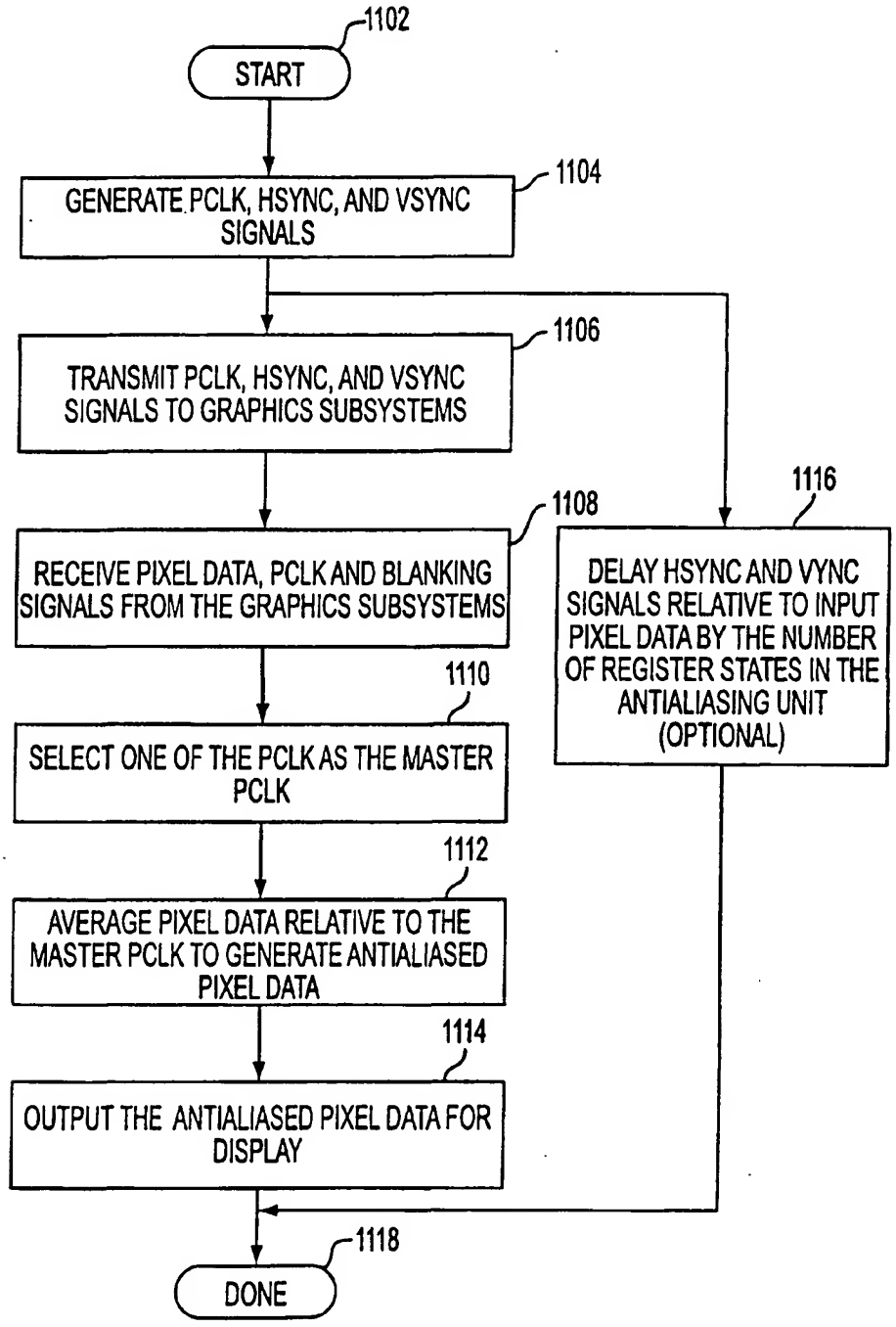

1102
START
1104
GENERATE PCLK, HSYNC, AND VSYNC SIGNALS
1106
TRANSMIT PCLK, HSYNC, AND VSYNC SIGNALS TO GRAPHICS SUBSYSTEMS
1108
RECEIVE PIXEL DATA, PCLK AND BLANKING SIGNALS FROM THE GRAPHICS SUBSYSTEMS
1110
SELECT ONE OF THE PCLK AS THE MASTER PCLK
1112
AVERAGE PIXEL DATA RELATIVE TO THE MASTER PCLK TO GENERATE ANTIALIASED PIXEL DATA
1114
OUTPUT THE ANTIALIASED PIXEL DATA FOR DISPLAY
1116
DELAY HSYNC AND VYNC SIGNALS RELATIVE TO INPUT PIXEL DATA BY THE NUMBER OF REGISTER STATES IN THE ANTIALIASING UNIT (OPTIONAL)
1118
DONE

FIG. 11

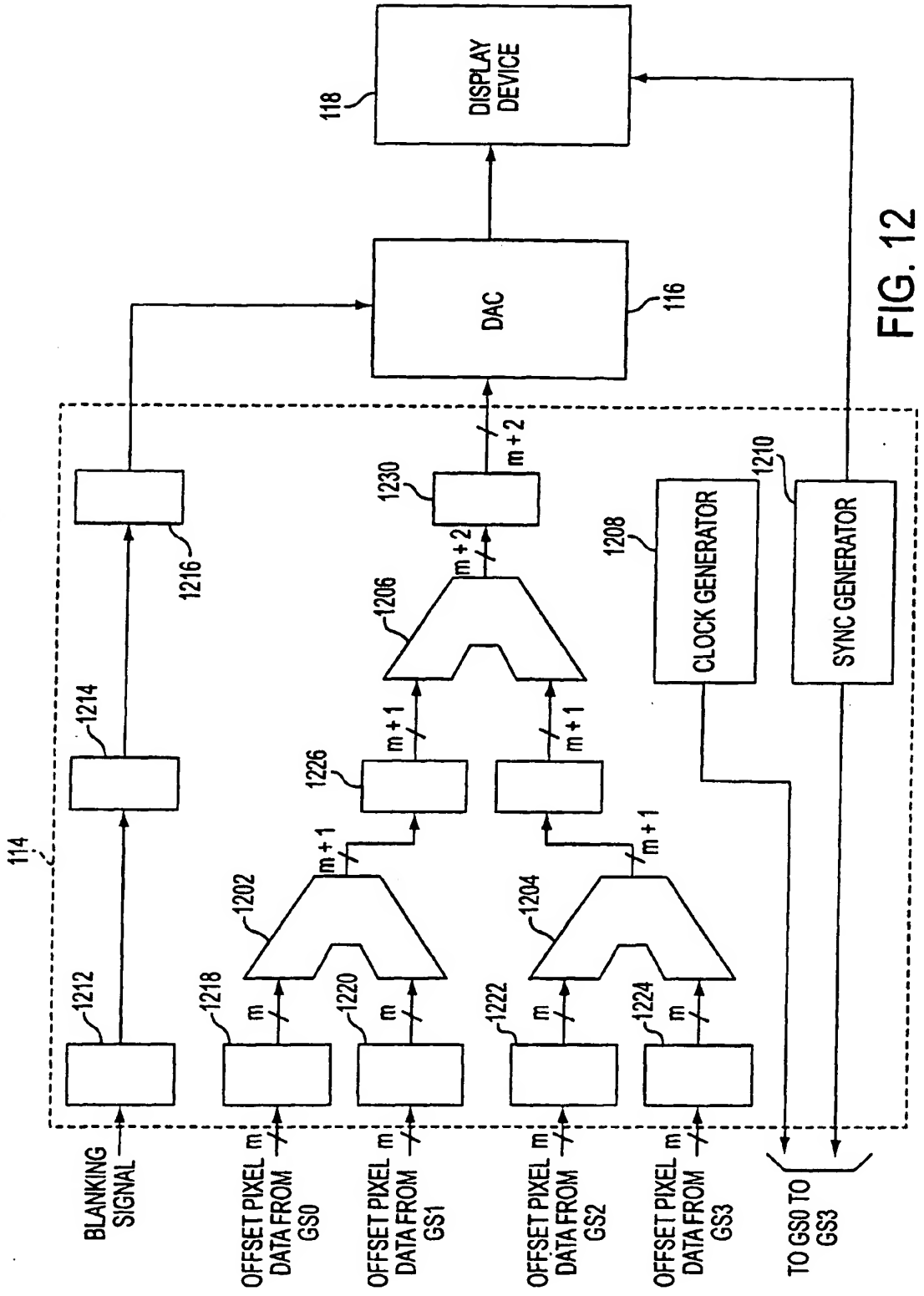

DISPLAY DEVICE
118
DAC
116
1216
1230
m + 2
1206
m + 2
1208
CLOCK GENERATOR
1210
SYNC GENERATOR
1214
m + 1
m + 1
1226
114
m + 1
m + 1
1202
1204
1212
1218
1220
1222
1224
m
m
m
m
BLANKING SIGNAL
OFFSET PIXEL DATA FROM GS0
OFFSET PIXEL DATA FROM GS1
OFFSET PIXEL DATA FROM GS2
OFFSET PIXEL DATA FROM GS3
TO GS0 TO GS3

FIG. 12

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/28003

A. CLASSIFICATION OF SUBJECT MATTER
IPC(7) :G06T 11/40
US CL :345/432, 136-138
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 345/432, 136-138

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS, EAST

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,748,178 A (DREWRY) 05 May 1998, entire document	1, 4-7, 9-55, 57-78, 80-100
Y	US 5,764,243 A (BALDWIN) 09 June 1998, entire document.	2-79, 81-100
Y	US 5,757,375 A (KAWASE) 26 May 1998, col. 4.	1, 80
Y,P	US 5,872,902 A (KUCHKUDA et al) 16 February 1999, entire document.	1-100
Y	US 5,798,770 A (BALDWIN) 25 August 1998, entire document	2-79, 81-100

☐ *Further documents are listed in the continuation of Box C.* ☐ *See patent family annex.*

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"B"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 23 MARCH 2000	Date of mailing of the international search report 17 APR 2000
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer MOTILEWA GOOD-JOHNSON Joni Hill Telephone No. (703) 308-3800

Form PCT/ISA/210 (second sheet) (July 1998)★